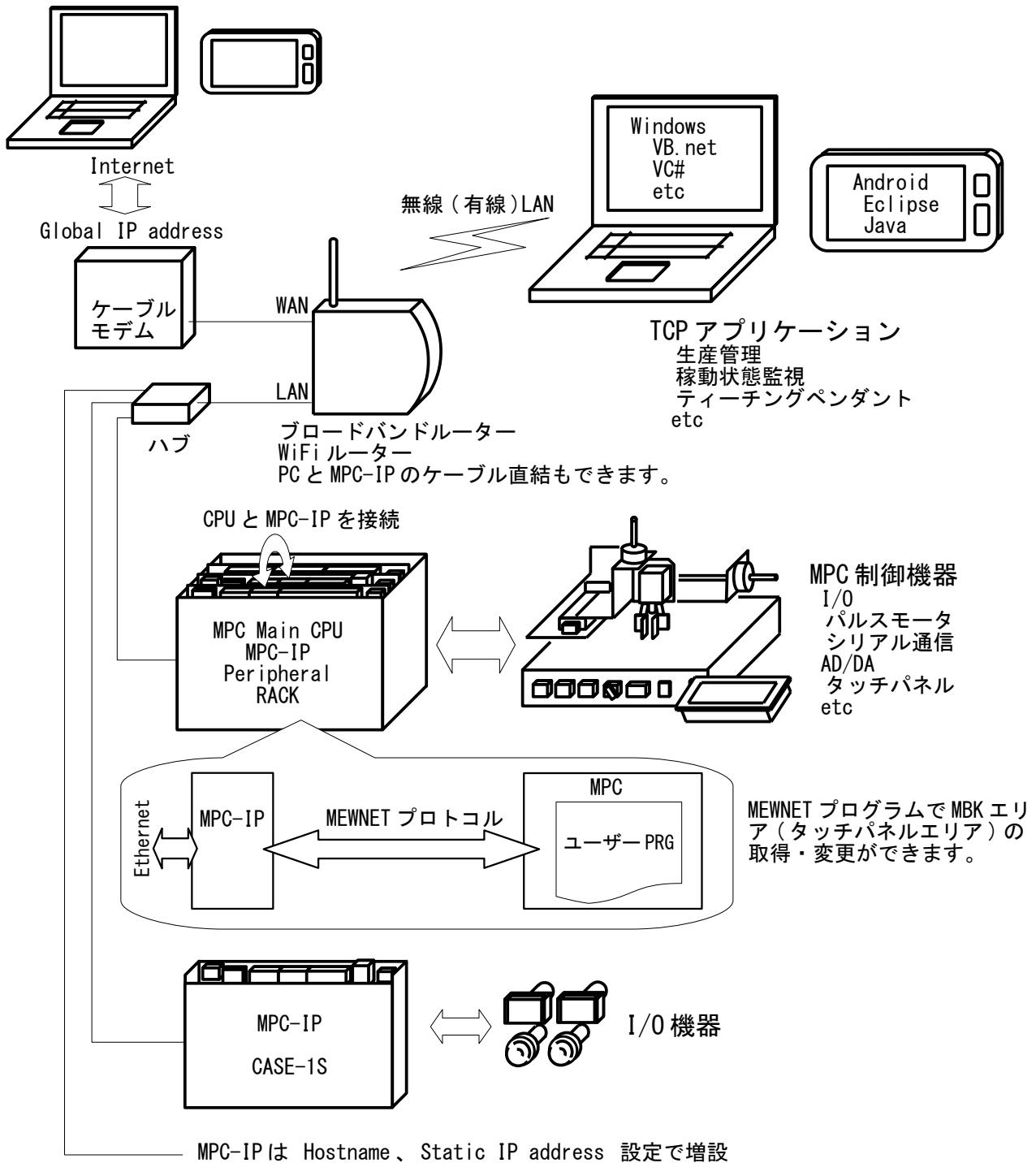


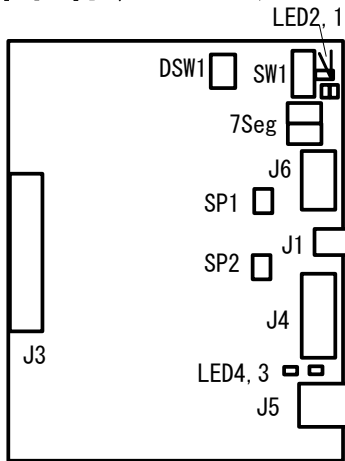
Technical Information		Ref No: ti2k-120905-1	Last Modify 160422
Title	MPC-IP マニュアル (Board model CEP-170A)		

システムイメージ

- MPC-IP は LAN・WAN から TCP/IP ソケット通信で MPC とのデータ授受を行う TCP/MPC プロトコルコンバータです。
- TCP アプリケーションは VB、VC、Eclipse などポピュラーな開発環境で作成できます。
- MPC-IP はフォトカプラ入力、TR オープンコレクタ出力(各 4 点)を備えているので単体でも簡単な I/O 制御ができます。
- TCP 通信は 1:1 です。複数のアプリが同時に同じ MPC-IP にアクセスすることはできません(一つのアプリで複数の MPC-IP にアクセスすることはできます)。



各部名称、ピン配列



- SW1 Slide Switch : MPC Prgram/MPC-IP Config 切替
- LED1 Connecting(点灯)
- LED2 Runnig(点滅)
- DSW1 Host Name の末尾番号設定
- 7Seg IP address 表示
- J6 PS10 ピン : MPC User Comm Port 接続
- SP1 ショートピン : J1 から給電する場合ショート
- SP2 ショートピン : J1 から給電する場合ショート
- J1 USB mini : MPC Program/MPC-IP Config (SW1 で切替)
- J4 PS16 ピン : 4 Inputs / 4 Outputs
- LED3 Network Access
- LED4 Network Connect(接続時消灯)
- J5 RJ-45 : Ethernet
- J3 バスコンネクタ (MPC CPU ボードから電源供給)

駆動電源 DC5V 250mA

※ 電源について

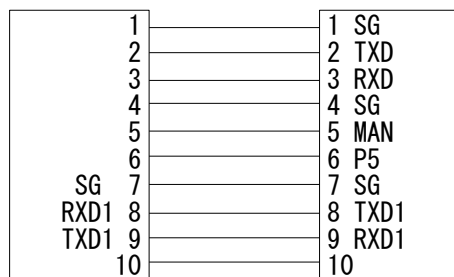
ラックまたはケースに収納して MPC-2000 シリーズ CPU ボードと併用する場合、駆動電源は J3 より供給されます。そのとき SP1,SP2 はオープンです。
 MPC-IP を単体で使用する場合は J1 に USB 電源を供給します。その場合は SP1、 SP2 をショートします。
 16 スロットラックで使用する場合は内部総消費電流に注意してください。(特に MPC-1000)

※ J1 との接続について

J1 は USB シリアルコンバータです。弊社 USB シリアルコンバータ「USB-RS」と同じデバイスドライバのセットアップが必要です。

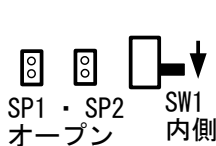
MPC-IP の J6 と各 MPC の J1 を 1:1 接続すると MPC-IP の J1 がプログラムポートになります。

MPC-IP J6

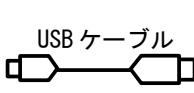


- MPC-1000 J1
- MPC-1200 J1
- MPC-2000 J1
- MPC-2200 J1

MPC の設定コマンド
 「MEWNET 38400 1」
 (タッチパネルの設定と同じコマンドです)

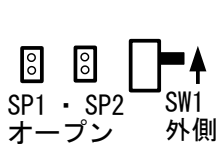


MPC-IP J1
 USB mini

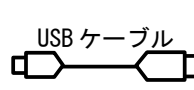


FTMW2K.EXE
 MPC Monitor

MPC のプログラムポートに接続され、FTMW の作業を行えます。



MPC-IP J1
 USB mini

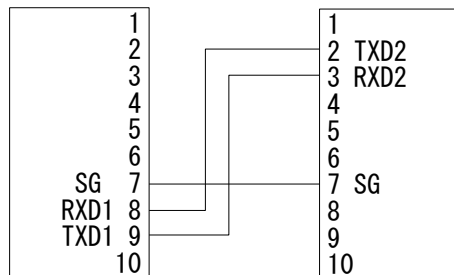


MPC1PCNF.EXE

MPC-IP の設定を行います。

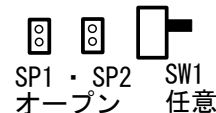
各 MPC の J1 以外に接続する場合は通常の RS-232C 機器と同様の 3 線接続です。

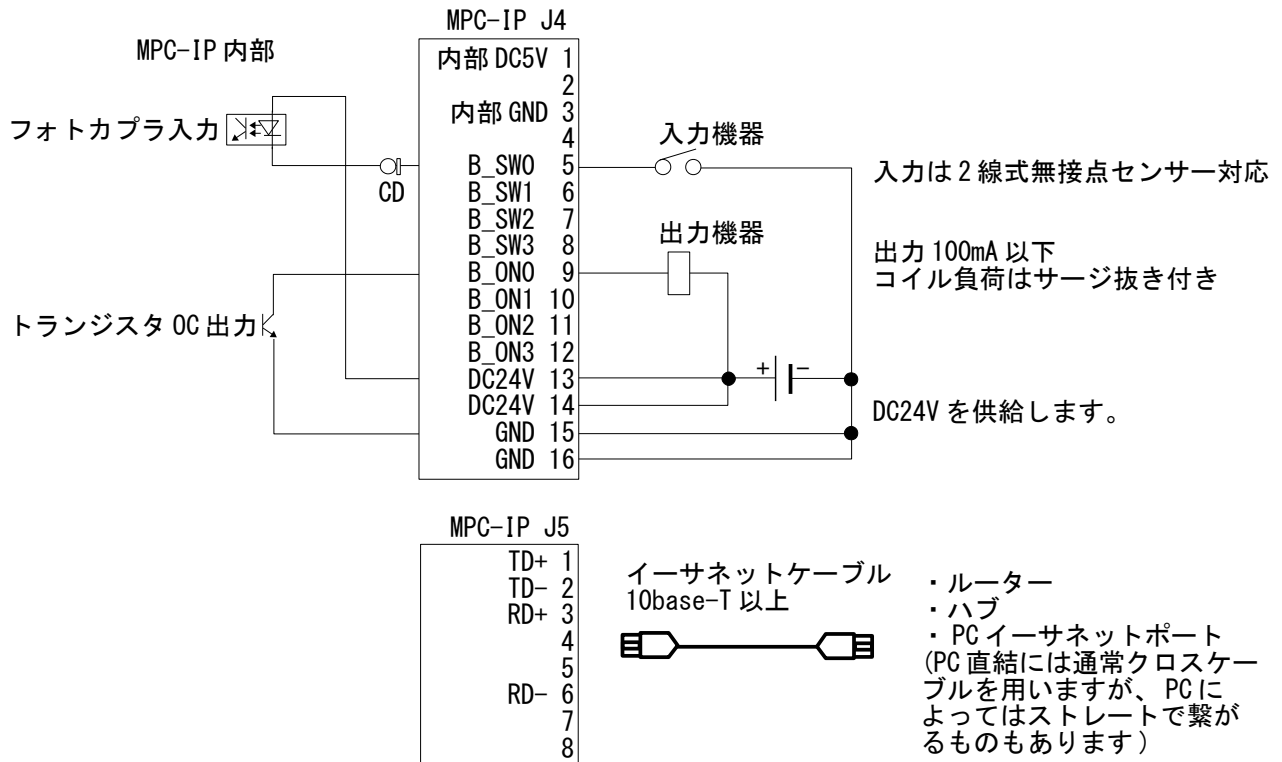
MPC-IP J6



- MPC-1000 J5
- MPC-1200 J6
- MPC-2000 J5
- MPC-2200 J5

MPC の設定コマンド
 MEWNET 38400 2

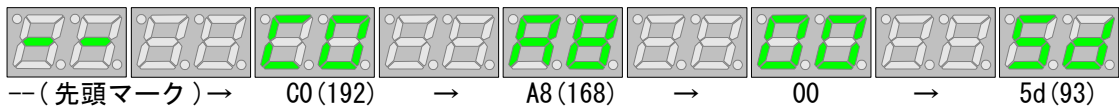




7Seg の表示

(a) IP address

クライアントアプリケーションからアクセスが無い状態(アイドル状態)のときは IP address を HEX で繰り返し表示します。例えば 192.168.0.93 の場合は次のようになります。



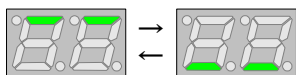
(b) IP address が変わったとき

DHCP から IP address を受け取りアドレスが変わったとき。(例) 192.168.0.99 から 192.168.0.93 に変わったとき。左上のドットも点灯します(1秒弱)。

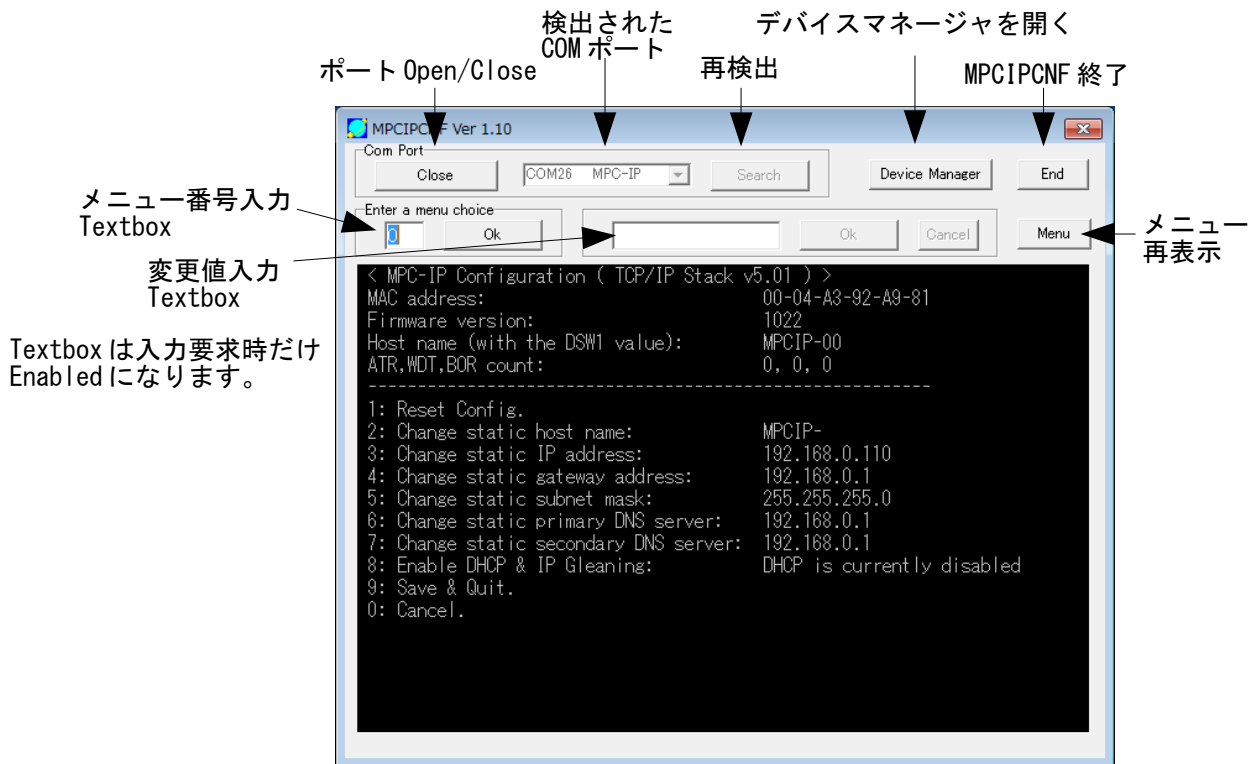


(c) クライアントアプリケーションからアクセスされたとき。

上下のセグメントが不規則に点滅します。10秒以上アプリケーションからアクセスが無いと(a)の表示に戻ります。



設定アプリケーション(MPCIPCNF.EXE)



Textbox は入力要求時だけ Enabled になります。

• 接続方法

(a) MPC ラック、ケースで使用する場合 (J3 コネクタ CPU 電源)
 SP1・SP2 をオープン、SW1 を外側にしてラック (ケース) に収納し、J1 と PC を USB ケーブルでつなぎ、MPC の電源を入れます。

(b) MPC-IP を単体で使用する場合 (J1 コネクタ USB 電源)
 SP1・SP2 をショート、SW1 を外側にし、J1 と PC を USB ケーブルでつなぎます。

※どちらの場合でも初回はデバイスドライバのセットアップが開始されます。

- (1) MPC-IP が稼動状態になっているのを確認してください (7 セグに IP アドレスを点滅表示) 。
- (2) MPCIPCNF. EXE を起動します。検出された MPC-IP のポート番号がコンボボックスに表示されます。
- (3) Com ポートを選択して「Open」ボタンを押します。
- (3) 正常に Com ポートが開かれると画面に「Com Port Opened」と表示され、直ぐにメニューが表示されます。
- (4) Textbox にメニュー番号、設定する値を入力します。

ヘッダー	意味
MAC address	MPC-IP ボード固有の物理アドレス
Firmware version	MPC-IP ボードのファームウェアバージョン
Host name (with the DSW1 value)	ホストネーム。下記メニュー 2 の static host name とボードの DSW1 の値 (2 桁) を組み合わせせた名前です。
RST, WDT, BOR count	アクセスタイムアウトリセット、ウォッチドッグタイムアウトリセット、ブラウンアウトリセットの回数。0 でなければ何らかの不具合があります。読込後 0 クリアされます。

メニュー	意味
1: Reset Config	MPC-IP の設定を初期状態に戻します。
2: Change static host name	ホスト名の固定部分を変更。 デフォルトは「MPCIP-」。
3: Change static IP address	IP アドレス固定値を変更。
4: Change static gateway address	Gateway アドレスを変更。
5: Change static subnet mask	サブネットマスクを変更。
6: Change static primary DNS server	Primary DNS Server アドレスを変更。
7: Change static secondary DNS server	Secondary DNS Server アドレスを変更。
8: Disable DHCP & IP Gleaning または Enable DHCP & IP Gleaning	DHCP & IP Gleaning 有効/無効を選択。 操作するごとに Disable/Enable が変わります。 Enable ならば、ネットワーク確立時に IP address が DHCP から与えられるアドレスに変わります。 Disable ならば メニュー 3 の static address で固定です。
9: Save & Quit	設定内容を EEPROM に保存して設定を終了します。
0: Cancel	設定内容を保存せずに設定を終了します。

- static host name を「MPCIP-」、ボードの DSW1 を 0 とすれば、Host name は“MPCIP-00”となります。Windows 環境なら 8:DHCP を Enable にすれば DHCP から与えられた IP address に関らず、Host name で接続できるはずです。
- Android からアクセス、LAN で IP アドレスを固定して使う、PC と直結する場合は、8:DHCP を Disable とし、3:static IP address を設定します。
- 4~7 は通常変更する必要はありません。
- 設定終了時

メニュー 9 または 0 で設定を終了し、MPC-IP は稼動状態に戻ります。

(上) メニュー 8:DHCP が Disable のとき。メニュー 3 で設定されたアドレスが表示されます。

(下) メニュー 8:DHCP が Enable のとき。IP address の初期値 (192.168.0.99) が表示されます。LAN に接続され、DHCP から IP アドレスが得られると IP アドレスが変わります。

MPCIPCNF.EXE はこの状態になったら閉じてください。

```

6: Change static primary DNS server: 192.168.0.1
7: Change static secondary DNS server: 192.168.0.1
8: Enable DHCP & IP Gleaning: DHCP is currently disabled
9: Save & Quit.
0: Cancel.

Configuration has completed.

Rebooted for SetConfig
192.168.0.110

```

```

7: Change static secondary DNS server: 192.168.0.1
8: Disable DHCP & IP Gleaning: DHCP is currently enabled
9: Save & Quit.
0: Cancel.

Configuration has completed.

Rebooted for SetConfig
192.168.0.99
192.168.0.84

```

MPC-IP クラスライブラリ [mpcip.dll]

- 参照の追加 (VB. net)

mpcip.dll を利用できるようにするためプロジェクトの参照に追加します。

(1) プロジェクトのフォルダ (例えば、Form1.vb と同じフォルダ) に mpcip.dll をコピーします。
(2) メインメニュー プロジェクト > 参照の追加 > 参照タブ で mpcip.dll を選択し OK を押し
ます。

- 関数の戻り値

1	正常終了
-1	MEWNET レスポンスの BCC エラー
-3	MPC コマンド、MEWNET 通信タイムアウト
-4	MPC コマンド 応答キャラクタ数オーバー
-64	MPC からの MEWNET BCC エラーコード
-256	一般的な例外

※関数自体の戻り値とパラメータ res は同じ値です。

※MPC はプログラム読込中や停止時 (Ctrl+A) などに通信が中断します。その時にアクセスすると -3 (通信タイムアウト) になります。res=-3 なら停止を回避するようにすればよいかも。

※MPC コマンドの文字数は送信受信とも 128 キャラクタ以下です (CRLF 含む)。とくに受信 (MPC からの
応答) キャラクタ数は変数の桁数や文字数などにより変動するので注意してください。

- クラス名

Public Class MPCIP

- 情報取得・設定関数

```
''' <summary>
''' DLLバージョン番号取得
''' </summary>
''' <param name="ver">バージョン文字列</param>
''' <param name="res">実行結果</param>
Public Function get_dll_ver (ByRef ver As String, ByRef res As Integer) As Integer

''' <summary>
''' ファームウェアバージョン取得
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="ver">バージョン文字列</param>
''' <param name="res">実行結果</param>
Public Function get_fw_ver (ByVal Strm As NetworkStream, ByRef ver As String, ByRef res As
Integer) As Integer

''' <summary>
''' FW IP address 取得
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="ip">IP address</param>
''' <param name="res">実行結果</param>
Public Function get_fw_IP (ByVal Strm As NetworkStream, ByRef ip As String, ByRef res As Integer)
As Integer

''' <summary>
''' Streamと文字列関連付け
''' IDはDLL内でエラー表示に使用します
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="ID">文字列</param>
''' <param name="res">実行結果</param>
Public Function set_stream_ID (ByVal Strm As NetworkStream, ByVal ID As String, ByRef res As
```

Integer) As Integer

```
''' <summary>
''' DLL message box En/Ds
''' エラー発生時、DLLのメッセージBOX表示の有無を選択します
''' (System例外は除く)
''' </summary>
''' <param name="v">Boolean</param>
''' <param name="res">実行結果</param>
```

Public Function dll_msgbox_enabled(ByVal v As Boolean, ByRef res As Integer) As Integer

```
''' <summary>
''' TCP Time out 設定
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="wtmout">WriteTimeOut</param>
''' <param name="rtmout">ReasTimeOut</param>
''' <param name="res">実行結果</param>
```

Public Function set_stream_timeout(ByVal Strm As NetworkStream, ByVal wtmout As Integer, ByVal rtmout As Integer, ByRef res As Integer) As Integer

```
''' <summary>
''' Sleep Time 設定
''' 関数終了時のSleep時間を設定します。初期値=2。
''' </summary>
''' <param name="Slp">Sleep</param>
''' <param name="res">実行結果</param>
```

Public Function set_sleep(ByVal slp As Integer, ByRef res As Integer) As Integer

```
''' <summary>
''' Read and Clear BOR, WDT, RST Counter
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="cnt">(上位バイト) BOR, WDT, RST (下位バイト)</param>
''' <param name="res">実行結果</param>
```

Public Function read_wdtrc(ByVal Strm As NetworkStream, ByRef cnt As Integer, ByRef res As Integer) As Integer

```
''' <summary>
''' Reset Access Watchdog Timer
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="res">実行結果</param>
```

Public Function reset_awdt(ByVal Strm As NetworkStream, ByRef res As Integer) As Integer

- MPC コマンド実行関数

```
''' <summary>
''' MPC Command
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="cmd">Command and Parameters</param>
''' <param name="ans">Answer from MPC</param>
''' <param name="res">実行結果</param>
```

Public Function mpc_cmd(ByVal Strm As NetworkStream, ByVal cmd As String, ByRef ans As String, ByRef res As Integer) As Integer

- MEWNET (MBK データエリア、I/O エリア アクセス)関数

```
''' <summary>
''' MEWNETタイムアウト設定(初期値3sec)
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="tm">設定値(msec)</param>
''' <param name="res">実行結果</param>
```

Public Function set_mewnet_tmout(ByVal Strm As NetworkStream, ByVal tm As Integer, ByRef res As Integer) As Integer

```
''' <summary>
```

```

''' MBKエリア書き込み
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="dt">書き込むデータ</param>
''' <param name="adr">MBKアドレス</param>
''' <param name="siz">サイズ</param>
''' <param name="res">実行結果</param>
Public Function mbk_write(ByVal Strm As NetworkStream, ByVal dt As Integer, ByVal adr As Integer, ByVal siz As Integer, ByRef res As Integer) As Integer

''' <summary>
''' MBKエリア読み込み
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="adr">MBKアドレス</param>
''' <param name="siz">サイズ</param>
''' <param name="dt">読み込んだデータ</param>
''' <param name="res">実行結果</param>
Public Function mbk_read(ByVal Strm As NetworkStream, ByVal adr As Integer, ByVal siz As Integer, ByRef dt As Integer, ByRef res As Integer) As Integer

''' <summary>
''' I/Oエリア パラレル出力
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="dt">出力するデータ</param>
''' <param name="adr">I/Oバンクアドレス</param>
''' <param name="siz">サイズ</param>
''' <param name="res">実行結果</param>
Public Function mbk_out(ByVal Strm As NetworkStream, ByVal dt As Integer, ByVal adr As Integer, ByVal siz As Integer, ByRef res As Integer) As Integer

''' <summary>
''' I/Oエリア パラレル入力
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="adr">I/Oバンクアドレス</param>
''' <param name="siz">サイズ</param>
''' <param name="dt">入力したデータ</param>
''' <param name="res">実行結果</param>
Public Function mbk_in(ByVal Strm As NetworkStream, ByVal adr As Integer, ByVal siz As Integer, ByRef dt As Integer, ByRef res As Integer) As Integer

''' <summary>
''' I/Oエリアビットon
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="adr">I/Oビットアドレス</param>
''' <param name="res">実行結果</param>
Public Function mbk_on(ByVal Strm As NetworkStream, ByVal adr As Integer, ByRef res As Integer) As Integer

''' <summary>
''' I/Oエリア ビットoff
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="adr">I/Oビットアドレス</param>
''' <param name="res">実行結果</param>
Public Function mbk_off(ByVal Strm As NetworkStream, ByVal adr As Integer, ByRef res As Integer) As Integer

''' <summary>
''' I/Oエリア ビット入力
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="adr">I/Oビットアドレス</param>
''' <param name="dt">入力状態 1=on、0=off</param>
''' <param name="res">実行結果</param>
Public Function mbk_sw(ByVal Strm As NetworkStream, ByVal adr As Integer, ByRef dt As Integer,

```


ByRef res As Integer) As Integer

- オンボード I/O 関数

```
''' <summary>
''' ボード上出力 bit on
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="bitn">bit番号</param>
''' <param name="res">実行結果</param>
Public Function b_on(ByVal Strm As NetworkStream, ByVal bitn As Integer, ByRef res As Integer)
As Integer
```

```
''' <summary>
''' ボード上出力 bit off
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="bitn">bit番号</param>
''' <param name="res">実行結果</param>
Public Function b_off(ByVal Strm As NetworkStream, ByVal bitn As Integer, ByRef res As Integer)
As Integer
```

```
''' <summary>
''' ボード上出力 パラレルout
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="dt">出力データ</param>
''' <param name="res">実行結果</param>
Public Function b_out(ByVal Strm As NetworkStream, ByVal dt As Integer, ByRef res As Integer) As
Integer
```

```
''' <summary>
''' ボード上入力 bit入力
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="swn">bit番号</param>
''' <param name="dt">入力状態</param>
''' <param name="res">実行結果</param>
Public Function b_sw(ByVal Strm As NetworkStream, ByVal swn As Integer, ByRef dt As Integer,
ByRef res As Integer) As Integer
```

```
''' <summary>
''' ボード上入力 ラッチbit入力
''' 保持されている入力を Read and Reset します
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="swn">bit番号</param>
''' <param name="dt">入力状態</param>
''' <param name="res">実行結果</param>
Public Function b_lsw(ByVal Strm As NetworkStream, ByVal swn As Integer, ByRef dt As Integer,
ByRef res As Integer) As Integer
```

```
''' <summary>
''' ボード上入力 パラレルin
''' </summary>
''' <param name="Strm">Stream</param>
''' <param name="dt">入力状態</param>
''' <param name="res">実行結果</param>
Public Function b_in(ByVal Strm As NetworkStream, ByRef dt As Integer, ByRef res As Integer) As
Integer
```

- 定数

MBK エリアアクセスのサイズ指定

```
Public Size_Int As Integer = 2 ' signed 2 byte, MPCの ~Int 相当
Public Size_Wrd As Integer = 4 ' unsigned 2 byte, MPCの ~Wrd 相当
Public Size_Lng As Integer = 8 ' signed 4 byte, MPCの ~Lng 相当
```

MPCプログラムの宣言

MPCのプログラムに、MPC-IPと接続するRS-232Cポートの宣言を記述します。タッチパネルと同じコマンドです。

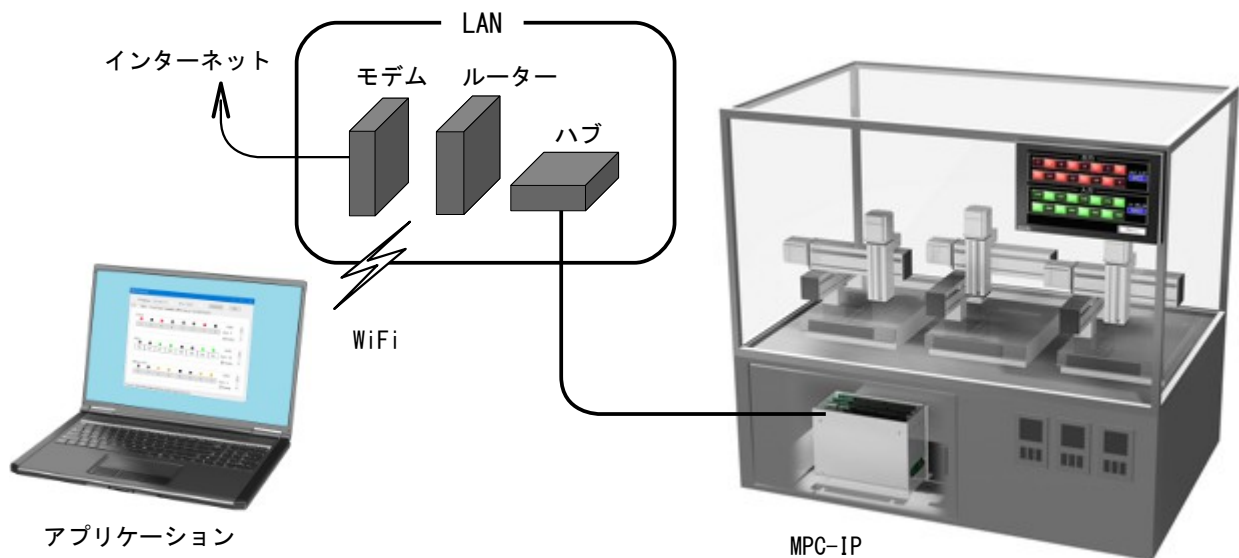
```
MEWNET 38400 1 ' Ch1 を使う場合
MEWNET 38400 2 ' Ch2 を使う場合
```

VB サンプルアプリケーション

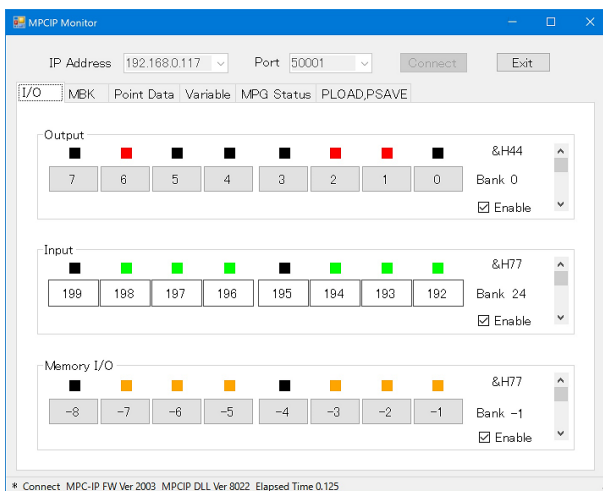
Visual Studio2013 の Visual Basicで作成したモニターです。
LAN上のPCから稼働状態を監視したり、MPCにログしたデータを取り込めます。
装置稼働中の接続・切断が可能です。

(プロジェクトファイルは「開発ツールダウンロード」
http://www.departonline.jp/acceleng/dev_utu.php の MPCIP_Monitor.zip です)

- イメージ



- アプリケーション画面
I/Oタブ

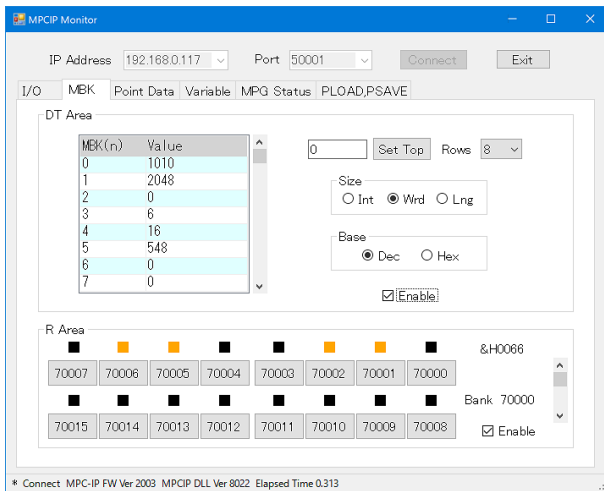


実出力のモニターと ON/OFF

実入力モニター

メモリー I/O のモニターと ON/OFF

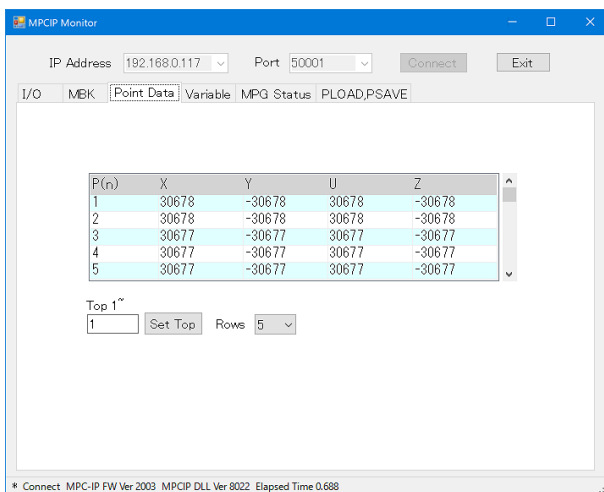
MBK タブ



MBK (DT) エリアのモニターと変更

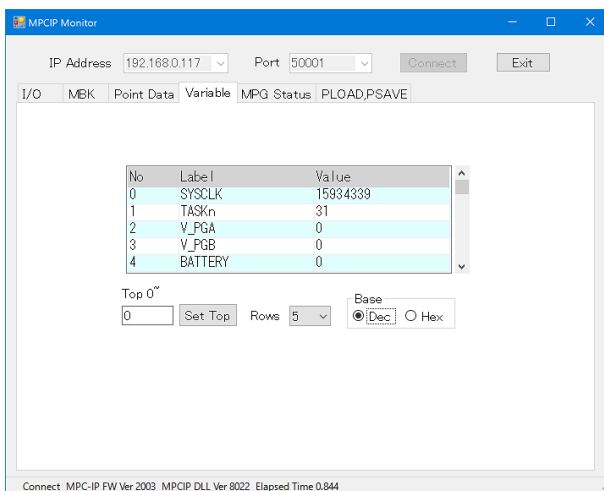
I/O (R) エリアのモニターと ON/OFF

Point Data タブ



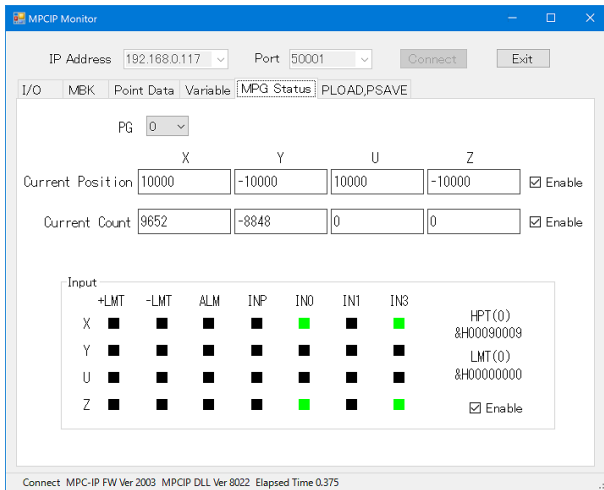
点データのモニターと変更

Variable タブ



変数のモニターと変更

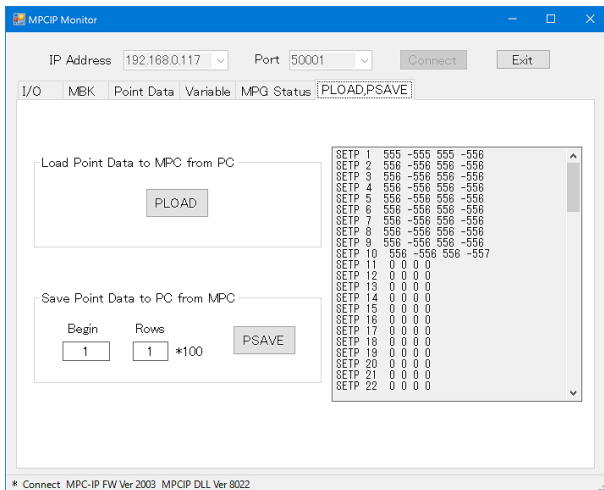
MPG Status タブ



現在座標値のモニター
カウンター値のモニター

入力のモニター

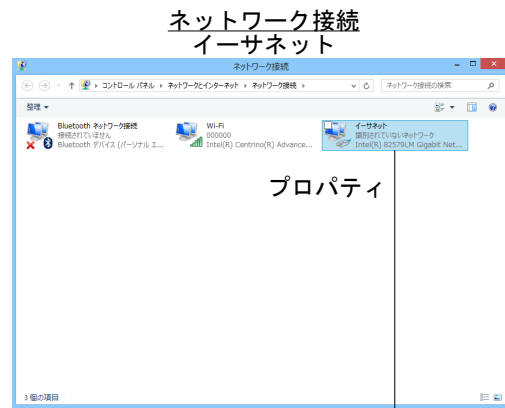
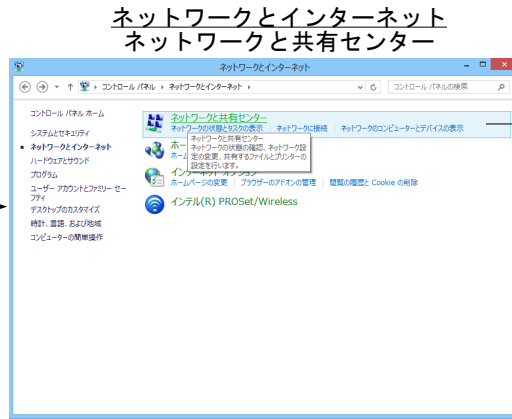
PLOAD, PSAVE タブ



PCから点データを読み込む

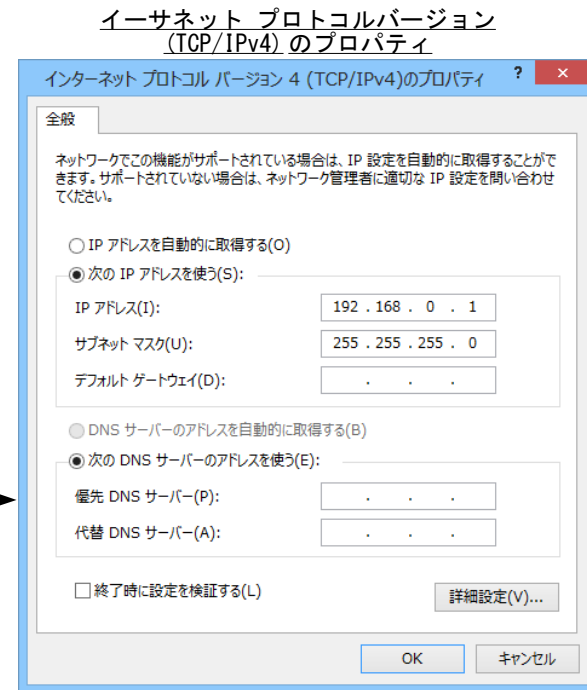
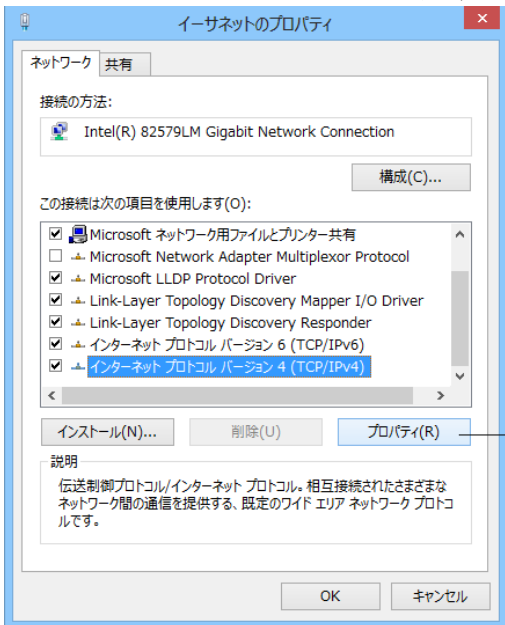
PCに点データを保存する

PC とケーブル直結時のイーサネットの設定



逆クリックでも入れます

イーサネットのプロパティ イーサネット プロトコルバージョン (TCP/IPv4)



PC の IP アドレスと MPC-IP を整合させます。

(例)
上記 IP アドレスなら MPC-IP の IP アドレスは
192.168.0.xxx になります。

Notes

ネット環境では使用機器、設定、経路、トラフィック、他のアプリの稼働などがレスポンスに影響します。リアルタイム性が要求される仕様では注意が必要です。下記は実験アプリケーションによるパフォーマンスの計測例です。

- Logitechのルータ（無線・インターネット）

Logitechのルータにはlogitecuserとlogitecgameuserの2つのSSIDがあり、およそ60~70秒間隔で停滞時間が発生します。

[環境]

PC : Panasonic Let's note、OS : Windows7 (x64)、アプリ : VB.net Express 製
ルータ : Logitech 「LAN-W300N/R」(据え置き型のルータ)
経路 : PC(無線) ⇄ 自宅ルータ ⇄ インターネット ⇄ 社内LAN ⇄ (有線)MPC-IP

応答時間 (観測時間 : 15分)

logitecuser (セキュリティの種類 : WPA2-パーソナル、暗号化の種類 : AES)

間欠停滞時間 : 7.322~14.339sec

logitecgameuser (セキュリティの種類 : 認証なし(オープンシステム)、暗号化の種類 : WEP)

間欠停滞時間 : 1.170sec

[環境]

PC : Panasonic Let's note、OS : Windows7 (x64)、アプリ : VB.net Express 製
ルータ : Logitech 「LAN-W300N/PR5B」(小型ハンディタイプの無線ルータ)
経路 : PC(無線) ⇄ ルータ ⇄ (有線)MPC-IP (WAN接続無し)

応答時間 (観測時間 : 15分)

logitecuser (セキュリティの種類 : WPA2-パーソナル、暗号化の種類 : AES)

通常0.031sec、最大7.020sec

logitecuser (セキュリティの種類 : WPA-パーソナル、暗号化の種類 : TKIP)

通常0.031sec、最大0.359sec

logitecgameuser (セキュリティの種類 : 認証なし(オープンシステム)、暗号化の種類 : WEP)

通常0.031sec、最大0.125sec

- BUFFALOのルータ (無線・LAN)

BUFFALOのルータでは間欠的な停滞は発生しないようです。

[環境]

PC : Panasonic Let's note、OS : Windows7 (x64)、アプリ : VB.net Express 製
ルータ : BUFFALO 「AirStation G」
セキュリティの種類 : 認証なし(オープンシステム)、暗号化の種類 : WEP
経路 : PC(無線) ⇄ LAN(ルータ) ⇄ (有線)MPC-IP

応答時間 (観測時間 : 15分)

通常0.031sec、最大0.640sec

- BUFFALOのルータ (有線・LAN)

BUFFALOのルータへ有線接続。

[環境]

PC : Dell Vostro(デスクトップ)、OS : Windows7 (x86)、アプリ : VB.net Express 製
ルータ : BUFFALO 「AirStation G」
経路 : PC(有線) ⇄ ハブ → LAN(ルータ) ⇄ (有線)MPC-IP

応答時間 (観測時間 : 15分)

通常0.020sec、最大0.072sec (IEなどに大きく影響されるようです)